REMARKS

Claims 1 – 43 are pending and stand rejected by the Office Action. The Office Action objected to claim 37 for an informality. Claim 37 has been amended to correct the informality, but has not been amended to overcome prior art or for any other reason substantially related to patentability. Claims 1 – 36 and 38 – 43 have not been amended.

§103(a) Rejections

The Office Action has rejected claims 1 – 9, 15 – 24, and 32 – 41 under 35 USC §103(a) as being unpatentable over Mark Allen Weiss's Data Structure and Algorithm Analysis in C++ Second Edition © 1999 (the Weiss reference) in view of the European Patent EP 0366585 B1 granted to Michael Edward Arnold (the Arnold reference). The Office Action has also rejected claims 10 – 14, 25 – 31, and 43 as being unpatentable over the Arnold reference in view of Douglas Comer's Operating System Design: The Xinu Approach © 1984 (the Comer reference).

Applicant respectfully traverses all rejections with respect to claims 1 – 43. Neither the combination of the Arnold reference with the Weiss reference nor the combination of the Arnold reference with the Comer reference teaches Applicant's claimed invention at least because **none of the references disclose** a double compare and swap operation or multi-way compare and swap operation to both update an array and return an indication of the state of the array as found in Applicant's claims.

As a preliminary matter, it is important to note that the Weiss, Arnold, and Comer references simply do not teach the subject matter the Office Action alleges they teach. First, the Weiss reference does not disclose a method or structure suitable for managing access to an array susceptible to concurrent operations thereon. The techniques of the Weiss reference do not even contemplate concurrent operations. Accordingly, and specifically, the techniques of the Weiss reference **do not include any** operation to atomically update any two quantities, let alone any operation that returns an indication of an array's state (see e.g., claim 1). As a result, the Office Action's interpretation of the Weiss reference as evidenced in paragraphs 5, 11, 16, 26, 28, 29, 31, and 36 of the Office Action is simply in error. The Weiss reference does not disclose an

atomic update operation. Furthermore, the non-existent atomic update operation cannot be said to return an indication of any sort, let alone the indications recited in various of the claims.

Second, contrary to the Office Action's statements (see paragraphs 6, 12, 18, 23, 27, 30, 32, and 37 of the Office Action) the Arnold reference does not disclose or suggest a DCAS or multi-way compare and swap, and the Office Action's assertion to the contrary is simply in error.

Third, and again contrary to the Office Action's statements, the Comer reference does not disclose or suggest detecting a boundary condition state of a double ended queue data structure, let alone disambiguating between a retry state and a boundary condition state based on an indication returned by a DCAS or multi-way compare and swap.

Each of the errant factual premises summarized above is rather fundamental and all rejections are unsustainable based on one or more of the observations above. We now turn more specifically to language of individual rejected claims.

The Weiss reference and the Arnold reference

Neither the Weiss reference nor the Arnold reference discloses or suggests "a double compare and swap (DCAS) to atomically update...and returning from the DCAS, on failure thereof, an indication by which...state of the array is detectable" as found in claims 1 and 6, "a double compare and swap (DCAS) to interrogate instantaneous values...for a signature indicative of ...state of the array...to atomically update" as found in claims 15 and 23, "a double compare and swap (DCAS) to atomically update a corresponding one, but not both, of the left and right indices L and R and an element of the contiguous array adjacent to the contiguous array element identified thereby" as found in claim 32, "a double compare and swap (DCAS) operation to atomically update...the contiguous array while returning on failure, an indication by which an empty state of the contiguous array is detected" as found in claim 36, or "a double compare and swap (DCAS) to atomically update...responsive to a corresponding boundary condition state of the concurrent object" as found in claim 40.

The Weiss reference **never discloses** atomically updating an array. The examples provided by the Weiss reference and cited by the Office Action include functions for performing pop and push operations. In the Weiss reference, specific functions are called to determine state

of an array (i.e., whether the array is full or empty), before an attempt to perform a pop or push on the array. After state of the array is determined from these functions, a series of operations are performed to manipulate the array. In contrast, the DCAS disclosed in Applicant's claims modifies an array and returns an indication of the state of the array if the attempt fails. An atomic operation as disclosed in Applicant's claims is neither disclosed nor suggested by the Weiss reference.

In addition, the Office Action mistakenly states that the Arnold reference discloses a double compare and swap (DCAS) operation and a multi-way compare and swap operation. The Arnold reference discloses a compare and swap (CAS) operation and a compare double and swap (CDS) operation as being "used in multi-programming and multi-processing environments to serialize access to counters, flags, control words and other common storage areas" (p.2, lines 51 – 52), but **never discloses** use of a DCAS operation or a multi-way compare and swap operation.

A double compare and swap operation and a CDS operation perform different operations. The DCAS operation performs at least two different compares and either performs two different assignments or does not perform the assignments, depending on the result of the compares. In contrast, a CDS operation performs a single compare, and one of two assignments, depending on the result of the single compare. The same functional differences exist between the multi-way compare and swap operation and the CDS operation.

In short, the CDS operation disclosed in the Arnold reference is different than the DCAS operation included in the limitations of independent claims 1, 6, 15, 23, 25, 32, 36, 40, and 43 and the multi-way compare and swap operation included in the limitation of independent claim 10.

In summary, neither the Weiss reference nor the Arnold reference disclose or suggest taken alone or in combination, use, as claimed, of atomic operations, such as a DCAS or a multi-way compare and swap. Accordingly, claims 1, 6, 15, 23, 32, 36, and 40 are allowable and a Notice of Allowance to that effect is respectfully requested.

The Arnold reference and the Comer reference

Neither the Arnold reference nor the Comer reference discloses or suggests "a first multi-way compare and swap... to disambiguate a retry state and a boundary condition state of the double-ended data structure... a second multi-way compare and swap performing the access and, on failure thereof, returning an indication disambiguating a retry state and the boundary condition state of the double-ended data structure" as found in claim 10, "a double compare and swap (DCAS) to interrogate instantaneous values...that identified thereby for a signature indicative of...state...a DCAS to atomically update" as found in claims 15 and 23, "a double compare and swap (DCAS) to atomically update... if successful completion of one of the first and the second competing access operations results in a boundary condition state of the array the DCAS fails and returns an indication thereof" as found in claims 25, or "at least one double compare and swap (DCAS) operation to disambiguate a retry state and a boundary condition state of the array" as found in claim 43.

Neither the Arnold reference nor the Comer reference disclose operations that manipulate an array and provide indication of the array's state. The Arnold reference specifically discloses separate operations to test pointers for determining state of an array (p. 9, lines 8 – 11, and lines 29 – 32 of the Arnold reference). **The CAL and CDS operations disclosed in the Arnold reference do not provide any indication of the state of an array**. The Comer reference discloses Boolean functions for testing states of an array, but these disclosed functions only determine the state of an array and do not update an array (p.45 of the Comer reference).

As previously discussed, the Arnold reference does not disclose or suggest a DCAS operation or a multi-way compare and swap operation.

Neither the Arnold reference nor the Comer reference standing alone or in combination teach or suggest Applicant's claimed invention. For at least the reasons stated above, Applicant respectfully submits that Applicant's independent claims 10, 15, 23, 25, and 43 are allowable, and a Notice of Allowance to that effect is respectfully requested.

OFFICIAL PATENT

Dependent Claims

Applicant respectfully traverses all rejections with respect to the dependent claims. All of the dependent claims are dependent on corresponding ones of the above allowable independent claims. Applicant respectfully submits that all of the dependent claims are allowable for at least the reasons discussed above.

Conclusion

In summary, claims 1 – 43 are in the case. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail, in an envelope addressed to Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

☒ facsimile transmitted to the US Patent and Trademark Office.

Steven R. Gilliam 8/7/2003
Steven R. Gilliam Date

EXPRESS MAIL LABEL: _____

Respectfully submitted,

Steven R. Gilliam, Reg. No. 51,734
Attorney for Applicant(s)
(512) 347-9030
(512) 347-9031 (fax)